Communications in Information and Systems
Volume 16, Number 4, 203–227, 2016

# A texture feature preserving image interpolation algorithm via gradient constraint

Hongwei Du, Yunfeng Zhang*, Fangxun Bao, Ping Wang, and Caiming Zhang

In this paper, a type of bivariate rational interpolation function is constructed for preserving image texture structure, which integrates polynomial function with rational function. On the basis of this model, an image interpolation algorithm for texture feature preserving is proposed. Firstly, isolines method is employed to divide image into smooth region and non-smooth region. Secondly, the smooth region and the non-smooth region are interpolated by polynomial model and rational model respectively. Finally, in order to preserve image texture direction, the pixel value of the interpolation point is determined by objective function that is constructed based on the gradient. Experimental results show that the proposed algorithm achieves good competitive performance compared with the state-of-the-art interpolation algorithms, especially in preserving image details and structure of edge.

## 1. Introduction

Image interpolation, which is the art of upscaling a low-resolution (LR) image to the high resolution (HR) version, has become a very active research field in image processing. It can be easily extended to diverse applications ranging from medical imaging, remote sensing, aviation, animation production and multimedia entertainment industries. Because of the relatively low computational complexity, traditional image interpolation methods such as bilinear, cubic spline and cubic convolution interpolations[1–3] are widely

*Research supported by the National Natural Science Foundation of China (Nos. 61373080, 61672018, U1609218, 61772309 and 61332015) and by the Nature Science Foundation of the Shandong Province of China (No. ZR2015AM007) and by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

used. But they often suffer from various forms of artifacts, such as blurring, ringing, jaggy edges and so on.

In general, the interpolation algorithm can be divided into discrete method and continuous method. The discrete method obtains the pixel value of the interpolated point by transforming the gray value of the known pixel. Carrato et al.[4] use the priori edge model to optimize the parameters of the interpolation operator, and then obtain the high quality interpolation image by interpolating the edge information on the basis of the original image. New edge-directed interpolation (NEDI)[5]makes use of geometric duality property to estimate HR covariance by LR covariance and interpolates the HR pixels using the estimated covariance. The method can effectively preserve the image edge structure. However, the interpolation window is fixed and can not be adjusted adaptively, which leads to distortion and deformation of the image texture details. Directional filtering and data fusion (DFDF)[6] is a new edge-guided non-linear interpolation algorithm. For each pixel to be interpolated, its neighborhood is divided into two observation subsets, and then the missing pixels are estimated from the two orthogonal directions. The algorithm can preserve the sharp edges of the image and reduce ring artifacts effectively. Soft-decision adaptive interpolation (SAI)[7] learns and adapts to varying scene structures by using a 2-D piecewise autoregressive model which interpolates the missing pixels in groups rather than one. This method maintains the spatial coherence of the interpolated images and achieves better subjective and objective effects in a wide range of scenarios. On the basis of SAI, robust soft-decision adaptive interpolation (RSAI)[8] improves the robustness of SAI by using weighted least squares estimation. Most of the adaptive interpolation methods based on discrete method cannot achieve the function that the image is magnified at any multiple. And these algorithms can not preserve the texture details of the image effectively, prone to some interpolation artifacts (blur, ringing, etc.). Further, combining with the machine learning method, Dong et al. [9] develop an image interpolation method in which nonlocal autoregressive modeling (NARM) is embedded in the sparse representation model. The method achieves outstanding results compared to the conventional edge-directed algorithms, however, it result in a significant cost of time.

Deep learning is an important branch of artificial intelligence research that has been given broad attention in recent years. Wang et al. combine the domain expertise of sparse coding and the merits of deep learning to propose a cascade of multiple SCNs (CSCN)[10]. This approach, motivated by the self-similarity based SR, uses a cascaded network for better flexibility in scaling factors and reduces artifacts for large scaling factors. However,

if the LR image does not contain sufficient repetitive patterns, then the algorithm tends to produce sharp edges rather than fine details. Dong et al. present a fully convolutional neural network for image super-resolution (SRCNN)[11]. This method can learn an end-to-end mapping between low- and high-resolution images and improve the image quality significantly. But these types of method that rely on an external data set have a considerable drawback: such methods are fixed and thus not adapted to the input image[12].

Compared with the discrete method, the continuous method can construct a continuous surface with the image sampling data. The method can scale the reconstructed image indefinitely. Because the natural image has nonlinear properties and the rational functions is a typical nonlinear model, the image interpolation algorithm based on the rational function has caused wide attention of scholars. Some image interpolation algorithms based on rational functions have been proposed in recent years[13–18]. These algorithms are applied to the image enlargement and enhancement. Liu et al.[19] propose a mixed weighted rational function image interpolation model whose weight coefficients are determined by the edge information of the sampling points. Zhao et al.[20] extend the 1-dimensional multi-node spline interpolation method to 2-dimensional, and then propose a multi-node spline image interpolation algorithm with compact support. It is superior to the traditional cubic convolution interpolation algorithm. In general, the image reconstructed by general rational function interpolation model has better visual effect and preserves the detail information of the image effectively. However, the general rational function interpolation model has been less than satisfactory in preserving structure of image edge.

In order to maintain the texture details and the structure of image edge, an image interpolation algorithm based on the $C^2$ continuous rational function is proposed. We first construct a novel type of bivariate rational interpolation function. With shape parameters varying, the function has different forms of expression. It combines the rational model and the polynomial model organically. Based on the constructed interpolation model, a method of region adaptive image interpolation is presented, which not only can adapt to image's local characters, but also reduce the time complexity. Further, we construct an objective function based on the gradient for preserving image texture direction.

The major contributions of this paper can be summarized in the following items: (1) A novel $C^2$ continuous bivariate rational interpolation model is constructed. (2) The image is divided into smooth region and non-smooth

region according to the regional feature. The smooth region and the non-smooth region are interpolated by polynomial model and rational model respectively. (3) An objective function based on image gradient is constructed to constrain the pixel values of interpolation points.

The remainder of this paper is organized as follows. In §2, we constructs a bivariate rational function interpolation model. In §3, the details of the proposed interpolation algorithm are described. The experimental results and analysis are presented to evaluate the effectiveness of the algorithm in §4.

## 2. Bivariate rational interpolation function

Based on our previous research work [13–16], we construct a novel $C^2$ continuous rational interpolation model. This section describes the construction process of the model.

Let $\Omega : [a, b; c, d]$ be the plane region, and $\{(x_i, y_j, f_{i,j}), i = 1, 2, \ldots, n; j = 1, 2, \ldots, m\}$ be a given set of data points, where $a = x_1 < x_2 < \cdots < x_n = b$ and $c = y_1 < y_2 < \cdots < y_m = d$ be the knot spacings, $f_{i,j} = f(x_i, y_j)$. $d_{i,j}$ and $e_{i,j}$ are chosen partial derivative value $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ at the knots $(x_i, y_j)$, respectively. Let $h_i = x_{i+1} - x_i$, and $l_j = y_{j+1} - y_j$, and for any point $(x, y) \in [x_i, x_{i+1}; y_j, y_{j+1}]$ in the $(x, y)-$plane. Let $\theta = \frac{x-x_i}{h_i}$ and $\eta = \frac{y-y_j}{l_j}$. Denoting

$$\triangle_{i,j}^{(x)} = \frac{f_{i+1,j} - f_{i,j}}{h_i}, \quad \triangle_{i,j}^{(y)} = \frac{f_{i,j+1} - f_{i,j}}{l_j}.$$

For each $y = y_j, j = 1, 2, \ldots, m$, construct the $x-$direction interpolation curve $P_{i,j}^*(x)$ in $[x_i, x_{i+1}]$, it is given by

$$P_{i,j}^* = \frac{p_{i,j}^*(x)}{q_{i,j}^*(x)}, \quad i = 1, 2, \ldots, n-1. \tag{1}$$

Where

$$\begin{aligned}
&p_{i,j}^*(x) = (1-\theta^3) f_{i,j} + \theta(1-\theta)^2 V_{i,j}^*(x) + \theta^3 f_{i+1,j} + \theta^2(1-\theta) W_{i,j}^*(x),\\
&q_{i,j}^*(x) = (1-\theta)^2 + \theta(1-\theta)\alpha_{i,j} + \theta^2,\\
&V_{i,j}^*(x) = \theta(1-(1-\theta)\alpha_{i,j})(f_{i+1,j} - f_{i,j} - h_i d_{i,j}) + h_i d_{i,j} + (\alpha_{i,j}+1) f_{i,j},\\
&W_{i,j}^*(x) = -(1-\theta)(1-\theta\alpha_{i.j})(f_{i+1,j} - f_{i,j} h_i d_{i+1,j}) + (\alpha_{i,j}+1) f_{i+1,j} - h_i d_{i,j},
\end{aligned}$$

with $\alpha_{i,j} > 0$. Let $P_{i,j}^*(x) = f_{i,j}$, $P_{i,j}^*(x_{i+1}) = f_{i+1,j}$, $P_{i,j}^{*'} = d_{i,j}$, $P_{i,j}^{*'}(x_{i+1}) = d_{i+1,j}$.

When

$$d_{i,j} = \frac{h_{i-1}\Delta_{i,j}^{(x)} + h_i\Delta_{i-1,j}^{(x)}}{h_{i-1}+h_i}, \quad i = 2,3,\ldots,n-1. \tag{2}$$

The interpolation function defined by (1) is $C^2$ continuous in the interval $[a,b]$, and which satisfies

$$P''(x_i) = \frac{2}{h_{i-1}+h_i}(\triangle_{i,j}^{(x)} - \triangle_{i-1,j}^{(x)}), \quad i = 2,3,\ldots,n-1.$$

The partial derivative of the node $x_1$ and $x_n$ are defined as follows:

$$\begin{aligned} d_{1,j} &= \Delta_{1,j}^{(x)} - \tfrac{h_1}{h_1+h_2}(\Delta_{2,j}^{(x)} - \Delta_{1,j}^{(x)}), \\ d_{n,j} &= \Delta_{n-1,j}^{(x)} - \tfrac{h_{n-1}}{h_{n-1}+h_{n-2}}(\Delta_{n-1,j}^{(x)} - \Delta_{n-2,j}^{(x)}). \end{aligned} \tag{3}$$

For each pair $(i,j), i = 1,2,\ldots,n-1$ and $j = 1,2,\ldots,m-1$, using the $x-$direction interpolation function $P_{x,y}^*(x)$ define the rational interpolation surface on $[x_i, x_{i+1}; y_j, y_{j+1}]$ as follows

$$P_{i,j}(x,y) = \frac{p_{i,j}(x,y)}{q_{i,j}(y)}, \tag{4}$$

where

$$\begin{aligned} &p_{i,j}(x,y) = (1-\eta)^3 P_{i,j}^*(x) + \eta(1-\eta)^2 V_{i,j} + \eta^2(1-\eta)W_{i,j} + \eta^3 P_{i,j+1}^*(x), \\ &q_{i,j}(y) = (1-\eta)^2 + \eta(1-\eta)\beta_{i,j} + \eta^2, \\ &V_{i,j} = (\beta_{i,j}+1)P_{i,j}^*(x) + l_j\phi_{i,j}(x) + \varphi_{i,j}(x,y), \\ &W_{i,j} = (\beta_{i,j}+1)P_{i,j+1}^*(x) - l_j\phi_{i,j+1}(x) + \psi_{i,j}(x,y), \\ &\phi_{i,s}(x) = (1-\theta)^3(1+4\theta+9\theta^2)e_{i,s} + \theta^3(6-8\theta+3\theta^2)e_{i+1,s}, s = j, j+1, \\ &\varphi_{i,j}(x) = (\eta - \eta(1-\eta)(\beta_{i,j}+1)) \times (P_{i,j+1}^*(x) - P_{i,j}^*(x) - l_j\phi_{i,j}(x)), \\ &\psi_{i,j}(x) = (1-\eta-\eta(1-\eta)(\beta_{i,j}+1)) \times (P_{i,j}^*(x) - P_{i,j+1}^*(x) + l_j\phi_{i,j}(x)), \end{aligned}$$

with $\beta_{i,j} > 0$. The interpolation function $P_{i,j}(x,y)$ defined by (4) is called the bivariate rational interpolation function and which satisfies

$$\begin{aligned} &P_{i,j}(x_r,y_s) = f(x_r,y_s), \quad \frac{\partial P_{i,j}(x_r,y_s)}{\partial x} = d_{r,s}, \\ &\frac{\partial P_{i,j}(x_r,y_s)}{\partial y} = e_{r,s}, \quad r = i, i+1; s = j, j+1. \end{aligned}$$

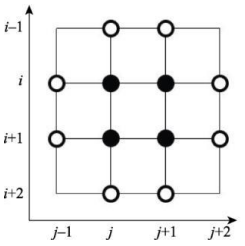



Figure 1: Rational interpolation model.

Fig. 1 shows the schematic diagram of the rational function interpolation model. The region surrounded by four black spots is the interpolation region. Based on this model, a surface patch is constructed by 12 points around the interpolation unit. Further, all interpolation surface patches of the whole image are connected to form an interpolated surface.

The interpolation model defined by (4) is determined by interpolation data and shape parameters. With shape parameters varying, the model has different forms of expression. In particular, when $\alpha_{i,j} = \beta_{i,j} = 1$, the interpolation model reduces a polynomial model.

## 3. Proposed algorithm

In this section, based on the constructed rational interpolation model above, a texture feature preserving image interpolation algorithm is proposed. First of all, the image is divided into the smooth region and the non-smooth region by using isolines method. Then, according to the regional features of the image, the rational model and the polynomial model are employed in image interpolation respectively. In final, a technique of interpolation point constraint in the non-smooth region is presented.

### 3.1. Adaptive region division

Regional division is a key step in image interpolation. The accuracy of regional division will directly affect the quality of interpolated images.

Isolines are the lines at the map that pass through the points with the same meanings of some quantitative indexes [21]. It has a wider range of applications in meteorology, geology, and so on. In this section, the isolines method is applied to the adaptive region division of the image.

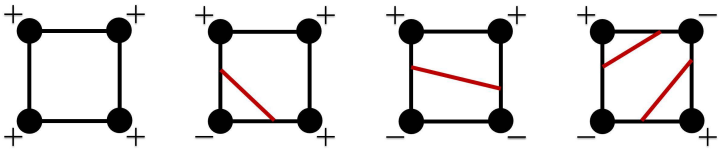


Figure 2: Schematic diagram of image region detection.

Isolines can describe the regional features of the image. In Fig. 1, a rectangle enclosed by $[i : i + 1; j : j + 1]$ (a rectangle surrounded by black spots) is defined as an interpolation unit. If there is at least an isoline in an interpolation unit, the region is defined as the non-smooth region, otherwise as the smooth region. The threshold of the image regional division is the mean value of the 12-pixels gray values in Fig. 1. So the threshold of the region detection varies with the movement of the interpolation unit. The adaptive detection threshold $\lambda$ for each patch is defined as

$$\lambda = \frac{\sum_{r=-1}^{2}\sum_{s=-1}^{2} f_{i+r,j+s} - (f_{i-1,j-1} + f_{i+2,j-1} + f_{i+2,j+2})}{12}.$$

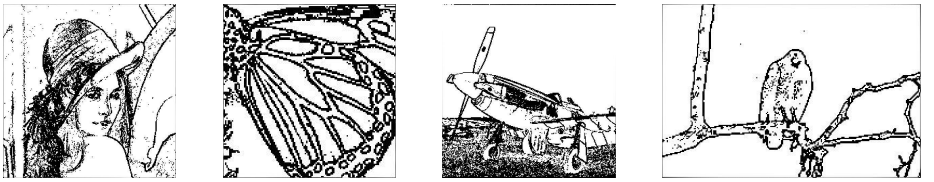
The difference between the four points' pixel values of the interpolation unit and $\lambda$ are denoted by $\Delta x_{m,n}, m, n = 0, 1$. If $\Delta x_{m,n}$ all be positive or negative, the region is defined as the smooth region, otherwise as the non-smooth region. Fig. 2 is a schematic diagram of image region detection. As shown in Fig. 2, if the four vertices have different symbols, the interpolation unit belongs to the non-smooth region. If the four vertices have the same symbols, the interpolation unit belongs to the smooth region.

Fig. 3 shows the detection results of Lena, Butterfly, Airplane and Hawk. It can be seen that the isolines method can effectively detect the texture and the boundary of image.

### 3.2. Image interpolation

The image is divided into the smooth region and the non-smooth region by using isolines method. The smooth region is interpolated by polynomial model, and the non-smooth region is interpolated by rational model.

Given a $m \times n$ image $I_{m,n}$, the pixel value of $i$ rows and $j$ columns in $I_{m,n}$ is denoted by $f_{i,j}(0 \le i \le m - 1, 0 \le j \le n - 1)$, and the pixel coordinates are $(i, j)$. In Fig. 4, the region surrounded by the four black dots is the



(a) (b) (c) (d)

Figure 3: (a)Texture region detection result of Lena. (b)Texture region detection result of Butterfly. (c)Texture region detection result of Airplane. (d)Texture region detection result of Hawk.

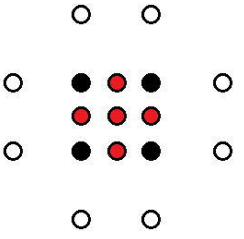


Figure 4: Image interpolation schematic diagram.

region to be interpolated. An interpolation surface patch of four black dots is constructed based on 12 pixels around the interpolation unit. Thus the pixel values of red pots can be obtained. The red dots are the interpolated pixels.

Due to the general rational function is $C^1$ continuous, the image interpolation models based on the rational function have higher fidelity in the smooth region and non-smooth region, however jaggy artifact are often present in the edges of the image. The reason is that the smoothness of the rational function model has an important influence on the quality of image interpolation. We construct the $C^2$ continuous rational interpolation model, which inherits the advantages of the general rational function in the non-smooth region, and reduces the appearance of the jagged edge of the interpolated image to some extent.

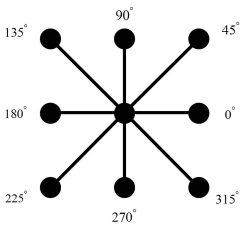



Figure 5: Schematic diagram of image texture orientation division.

### 3.3. Interpolation point constraints based on image gradient

In order to restore the texture details of the image better, we propose an objective function that constrains the interpolation point based on the image gradient. The gradient of the interpolation unit is obtained according to the isotropic sobel operator, and the texture direction is determined. Then, we construct an objective function that is used to calculate the weight coefficient of interpolation unit. Finally, we calculate the weight of the correlation point, and determine the pixel value of the interpolation point by convolution.

**3.3.1. Texture direction determine.** As shown in Fig. 5, the texture of the image interpolation unit is divided into four directions: horizontal, vertical and diagonal. The image gradient is calculated by the isotropic sobel operator, and the texture direction is determined according to the image gradient.

The isotropic sobel operator is used to detect the horizontal and vertical textures of the image [22]. On the basis of the isotropic sobel operator, the horizontal and vertical texture detection templates are rotated by 45 degrees to obtain the diagonal texture detection template. The $3 \times 3$ texture detection templates of the isotropic sobel operator is represented by $M_1, M_2, M_3, M_4$. Let $M_1$, $M_2$ be the isotropic sobel operator template of vertical, horizontal texture. Let $M_3$, $M_4$ be the isotropic sobel operator template of diagonal texture. If we define $A$ as the source image of the interpolation unit, $G_{x1}$ and $G_{y1}$ are two images which at each point contain the horizontal and vertical derivative approximations respectively, and $G_{x2}$ and $G_{y2}$ are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows:

$$G_{x1} = M_1 * A, \quad G_{y1} = M_2 * A,$$
$$G_{x2} = M_3 * A, \quad G_{y2} = M_4 * A.$$

where $*$ here denotes the 2-dimensional signal processing convolution operation.

Let $G_1$ be the gradient magnitude of vertical, horizontal texture. Let $G_2$ be the gradient magnitude of diagonal texture. $\theta_1, \theta_2$ is the corresponding image gradient's direction respectively. Based on the above analysis, the gradient of each image patch can be obtained. The calculation formulas of image gradient are as follows:

$$G_1 = \sqrt{G_{x1}^2 + G_{y1}^2}, \quad G_2 = \sqrt{G_{x2}^2 + G_{y2}^2},$$
$$\theta_1 = arctan(\tfrac{G_{y1}}{G_{x1}}), \quad \theta_2 = arctan(\tfrac{G_{y2}}{G_{x2}}).$$

For each interpolation unit of the image, the texture direction is determined by comparing the image gradients in four directions. $G_1 >= G_2$ indicates that the texture is in the vertical or horizontal direction, otherwise the texture is in the diagonal direction. Fig. 6 is a schematic diagram for judging the direction of image texture. Fig. 6(a) is a schematic diagram for judging the horizontal or vertical direction texture. If the image gradient direction $\theta_1$ is between $-45$ degrees and 45 degrees(Fig. 6(a), $\theta_1$ is in the red area), it is a vertical texture, otherwise it is a horizontal texture. Fig. 6(b) is a schematic diagram of the image diagonal direction texture. If the image gradient direction $\theta_2$ is between 0 degrees and 90 degrees(Fig. 6(b), $\theta_2$ is in the blue area), the texture is in the 135 degrees-direction, otherwise it is the diagonal texture.

**3.3.2. Interpolation point conditional constraint.** In order to preserve the texture features of the image well, an objective function that constraints the interpolation pixels based on image gradient is constructed. The objective function maintains the texture direction of the reconstructed image consistent with the sampled image by constraining the interpolation pixels.

An image gradient is a directional change in the intensity or color in an image. In an interpolation unit, the change of the pixel values may affect the change of texture direction. In order to keep the texture direction of the sampled image consistent with the reconstructed image, we construct the objective function which is used to constrain the pixel values of interpolation

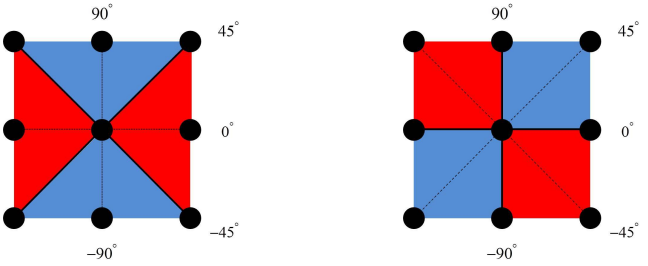



Figure 6: (a)Horizontal, vertical direction texture judgment. (b)Diagonal direction texture judgment.

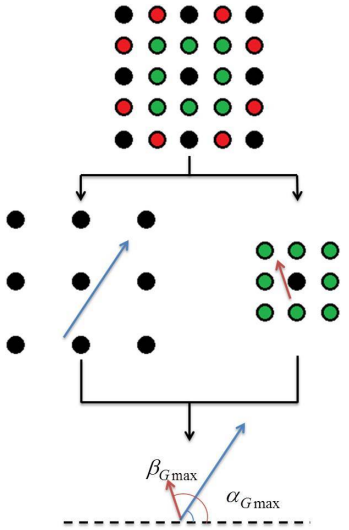



Figure 7: Schematic diagram of interpolation pixels constraint based on the gradient.

points. The objective function is given by

$$F = min(||\alpha_{Gmax} - \beta_{Gmax}||).$$

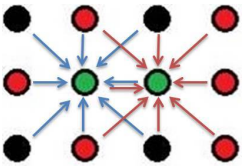


Figure 8: Schematic diagram of interpolated pixel mixed weighted.

Fig. 7 shows the schematic diagram of interpolation pixels constraint based on the image gradient. $\alpha_{Gmax}$ is the direction of the maximum gradient of the sampling pixel in an interpolation unit, it is fixed. $\beta_{Gmax}$ is the direction of the maximum gradient of the interpolated pixel in an interpolation unit. When the values of $\alpha_{Gmax}$ and $\beta_{Gmax}$ remain in the same, we leave the pixel values of interpolation points unchanged, otherwise we constrain the pixel values of the interpolation points. Based on the proposed objective function, in the next section, we give the approximate method for calculating the weight coefficient.

**3.3.3. Weight calculation.** The texture direction of the interpolated patch is determined according to the image gradient. We calculate the weight of the correlation point for the interpolation unit of the different texture direction, and determine the definitive pixel value of the interpolation point.

Fig. 8 is schematic diagram of interpolated pixel mixed weighted. Black dot is the sampling pixel, the red dot is the pre-interpolated pixels, and the green dot is the interpolated pixels.

In the $[i, j]$ interpolation unit, let $l_{r,s}$ be the connection between the pixel$[i + r, j + s](r, s = 0, 1, 2)$ and the center pixel$[i + 1, j + 1]$. let $\theta_{i+r,j+s}$, $(\theta_{i+r,j+s} \in [0, \pi/2])$ be the angle between $l_{r,s}$ and the texture direction. Let $\varpi_{i+r,j+s}, (r, s = 0, 1, 2)$ be the weighting factor of $[i + r, j + s]$pixel; this is given by

$$\varpi_{i+r,j+s} = \frac{\pi}{2 \times \theta_{i+r,j+s}}. \tag{5}$$

In (5), the range of $\varpi_{i+r,j+s}$ is $[1, +\infty]$. When $\theta_{i+r,j+s}$ is 0, the value of $\varpi_{i+r,j+s}$ is $+\infty$, it can not be calculated. A large number of experiments have shown that when the range of $\varpi_{i+r,j+s}$ is mapped to $[0, 2]$, the results are satisfactory. The mapping function is defined as

$$g : \varpi_{i+r,j+s} \to w_{i+r,j+s}, w_{i+r,j+s} \in [0, 2].$$

According to the weighting factor, we can obtain the weight coefficient matrix of each interpolation unit by convolution. The weight coefficient matrix on the interpolation unit $[i, j]$ is defined as

$$\begin{pmatrix} w_{i,j} & w_{i,j+1} & w_{i,j+2} \\ w_{i+1,j} & w_{i+1,j+1} & w_{i+1,j+2} \\ w_{i+2,j} & w_{i+2,j+1} & w_{i+2,j+2} \end{pmatrix}$$

Let $w_{i+r,j+s}$ be the weight coefficient at the pixel$[i + r, j + s]$ on the interpolation unit$[i, j]$, and let $f_{i+r,j+s}$ be the pixel value at the point$[i + r, j + s]$ on the interpolating unit $[i, j]$. The interpolation unit is convoluted with the weight coefficient matrix to obtain the interpolated pixel value. The term $f_{i+1,j+1}$ on the interpolation unit$[i, j]$ is defined as

$$f_{i+1,j+1} = \frac{\sum_{r=0}^{2}\sum_{s=0}^{2} f_{i+r,j+s} \times w_{i+r,j+s}}{\sum_{r=0}^{2}\sum_{s=0}^{2} w_{i+r,j+s}} (r, s = 0, 1, 2).$$

## 4. Experimental results

Experiments are conducted to evaluate the effectiveness of the interpolation algorithm. The proposed algorithm is compared with linear interpolator Bicubic, as well as recent state-of-the-art methods including directional filtering and data fusion (DFDF) [6], robust soft decision adaptive interpolation (RSAI) [8], nonlocal autoregressive model (NARM) [9], the cascade of SCNs (CSCN) [10], Super-Resolution Convolutional Neural Network model (SRCNN)[11], image interpolation based on weighted and blended rational function (Liu's)[17], an edge−guided image interpolation method using Taylor series approximation (Lee's) [23].

In our experiments, Bicubic interpolation is performed with the MATLAB built-in function, and the source codes of other compared methods are kindly provided by their authors. The LR source image is obtained through down-sampling the original HR image by a factor of 2 (both horizontally and vertically, no pre-filtering).

Peak signal to noise ratio(PSNR) is an average quality measurement over the whole image. Structural similarity index(SSIM) is a visual assessment of the quality of the image. So we use PSNR and SSIM as metrics to measure the performance of these interpolation algorithms. Tables 1 and 2 present the PSNR and SSIM of various methods, respectively. The average PSNR

of the proposed method is 1.77, 0.3, 1.26, 0.31, 0.58, 0.92, 4.26 and 3.53 dB higher than that of the Bicubic, RSAI, DFDF, NARM, Liu's, Lee's, CSCN and SRCNN, respectively. The average SSIM of the proposed method is 0.0711, 0.007, 0.0096, 0.0058, 0.009, 0.0197, 0.1247 and 0.1053 higher than that of the Bicubic, RSAI, DFDF, NARM, Liu's, Lee's, CSCN and SRCNN, respectively.

Table 1: PSNR comparison of different metods

| | BICUBIC | RSAI | DFDF | NARM | Liu's | Lee's¡¡ | CSCN | SRCNN | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| Baboon | 21.60 | 22.93 | 22.25 | 22.56 | 22.46 | 22.03 | 18.56 | 19.87 | **23.04** |
| Barbara | 23.19 | 22.99 | 24.07 | 23.37 | 23.68 | 22.72 | 20.14 | 20.70 | **24.55** |
| Brick | 32.57 | 35.05 | 32.22 | 35.05 | 34.24 | 34.93 | 30.29 | 30.60 | **35.30** |
| Dollar | 18.08 | 18.96 | 18.67 | 18.77 | 18.70 | 18.30 | 15.91 | 16.50 | **19.06** |
| Fence | 19.38 | 20.59 | 20.96 | 21.39 | 21.19 | 20.60 | 17.64 | 17.69 | **21.40** |
| Metal | 13.68 | 14.47 | 14.35 | 13.72 | 14.17 | 13.30 | 11.13 | 12.27 | **14.69** |
| ParrotsRGB | 28.46 | 32.41 | 27.05 | **32.58** | 31.15 | 30.95 | 26.11 | 26.83 | 32.46 |
| Rail | 21.85 | 22.88 | 22.51 | 22.81 | 22.61 | 22.31 | 20.10 | 20.42 | **23.18** |
| Sky | 28.43 | 29.75 | 29.19 | **29.88** | 29.46 | 29.21 | 25.84 | 26.60 | 29.85 |
| LionRGB | 26.16 | 27.98 | 27.75 | 27.94 | 27.78 | 27.55 | 23.00 | 24.15 | **28.01** |
| Wall | 23.30 | 24.89 | 24.09 | 24.90 | 24.60 | 24.05 | 20.50 | 21.67 | **24.91** |
| Girl | 29.27 | 30.72 | 29.05 | 30.62 | 30.27 | 30.30 | 26.82 | 27.48 | **30.75** |
| Average | 23.83 | 25.30 | 24.34 | 25.29 | 25.02 | 24.68 | 21.34 | 22.07 | **25.60** |

Table 2: SSIM comparison of different metods

| | BICUBIC | RSAI | DFDF | NARM | Liu's | Lee's | CSCN | SRCNN | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| Baboon | 0.7684 | 0.8671 | 0.8556 | 0.8626 | 0.8575 | 0.8483 | 0.7107 | 0.7345 | **0.8697** |
| Barbara | 0.7956 | 0.8548 | 0.8714 | 0.8590 | 0.8621 | 0.8444 | 0.7492 | 0.7511 | **0.8754** |
| Brick | 0.9143 | 0.9600 | 0.9562 | 0.9550 | 0.9540 | 0.9614 | 0.8729 | 0.8781 | **0.9634** |
| Dollar | 0.7170 | 0.8022 | 0.8022 | 0.7971 | 0.7915 | 0.7786 | 0.6843 | 0.6925 | **0.8053** |
| Fence | 0.6223 | 0.7014 | 0.7140 | **0.7265** | 0.7116 | 0.6960 | 0.5657 | 0.5824 | 0.7188 |
| Metal | 0.4369 | 0.5129 | 0.4989 | 0.5102 | 0.5242 | 0.4653 | 0.3700 | 0.4101 | **0.5374** |
| ParrotsRGB | 0.8968 | **0.9312** | 0.9194 | 0.9306 | 0.9210 | 0.9225 | 0.8535 | 0.8836 | **0.9312** |
| Rail | 0.7358 | 0.7818 | 0.7779 | 0.7800 | 0.7696 | 0.7656 | 0.6594 | 0.6847 | **0.7831** |
| Sky | 0.8455 | 0.9093 | 0.9082 | 0.9105 | 0.9051 | 0.9012 | 0.7992 | 0.8141 | **0.9115** |
| LionRGB | 0.7762 | 0.8358 | 0.8267 | 0.8383 | 0.8376 | 0.8335 | 0.6819 | 0.7147 | **0.8389** |
| Wall | 0.7932 | 0.8833 | 0.8798 | 0.8863 | 0.8805 | 0.8715 | 0.7449 | 0.7746 | **0.8877** |
| Girl | 0.9337 | 0.9657 | 0.9629 | 0.9630 | 0.9658 | 0.9641 | 0.9002 | 0.9042 | **0.9667** |
| Average | 0.7696 | 0.8337 | 0.8311 | 0.8349 | 0.8317 | 0.8210 | 0.7160 | 0.7354 | **0.8407** |

It is well known that objective metric is not completely in accordance with subjective tests, the evaluation of the visual appearance of the interpolated images is required for different methods. In Figs. 9–22, we demonstrate the visual effect of the proposed method and other methods. From

the subjective quality comparisons shown in these figures, we can find that the proposed algorithm is superior to other algorithms in preserving image texture detail and edge structure.

Figs. 9–18 compare the visual effect of the texture detail of the reconstructing images. As shown in Figs. 10 and 12, compared with the other state-of-the-art methods, the proposed method is more appealing in preserving texture direction. BICUBIC suffers from blurry artifacts. The images generated by DFDF and Lee's produce noise. The results achieved by NARM, Liu's and DFDF are distorted. The images obtained by RSAI observe jagged edges. The texture details are not efficiently preserved in CSCN and SRCNN. From Figs. 16 and Fig. 18, we can see that NARM, DFDF, Liu's, CSCN lost detail information, RSAI, SRCNN and Lee's produce texture distortion and deformation. When the image have complex texture, it is not satisfactory with image restoration using Bicubic. Our approach is able to obtain vivid texture regions. Overall, our method has outstanding advantages in preserving image texture direction and detail.

Figs. 19–22 compare the visual effect of the edge structure of the reconstructing images. In Figs. 20 and 22, it can be clearly observed that the images reconstructed by Bicubic, RSAI, Liu's, Lee's, CSCN and SRCNN interpolator suffer from jaggy. The DFDF, RSAI and the proposed method are stable for most edge structure, while the results achieved by the proposed method seem more sharp and natural. Obviously, our proposed image interpolation algorithm delivers the best subjective results with clear texture detail, sharp edges and least amount of artifacts incurred.

Table 3: The run time comparison of different methods (unit: second).

| | BICUBIC | DFDF | RSAI | NARM | Liu's | Lee's | CSCN | SRCNN | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| Baboon | 0.49 | 12 | 197 | 1547 | 6.6 | 21 | 5.0 | 9.6 | 4.8 |
| Barbara | 0.35 | 12 | 202 | 1411 | 5.2 | 26 | 4.2 | 9.8 | 3.7 |
| Brick | 0.47 | 12 | 207 | 1550 | 6.4 | 11 | 4.2 | 9.9 | 3.7 |
| Dollar | 0.63 | 12 | 205 | 1526 | 5.1 | 20 | 4.0 | 10.0 | 3.7 |
| Fence | 0.27 | 2 | 26 | 139 | 0.8 | 3 | 2.3 | 1.4 | 0.5 |
| Metal | 0.74 | 12 | 191 | 1702 | 8.7 | 21 | 4.2 | 9.7 | 3.8 |
| ParrotsRGB | 0.65 | 9 | 176 | 1453 | 3.2 | 19 | 2.6 | 2.9 | 2.6 |
| Rail | 0.46 | 3 | 43 | 246 | 3.7 | 4 | 2.5 | 2.7 | 0.9 |
| Sky | 1.17 | 19 | 310 | 2408 | 12.8 | 24 | 5.4 | 15.7 | 5.5 |
| LionRGB | 0.57 | 10 | 206 | 1579 | 4.8 | 22 | 3.3 | 5.6 | 3.4 |
| Wall | 0.78 | 18 | 312 | 2496 | 10.5 | 24 | 5.2 | 15.4 | 5.5 |
| Girl | 0.44 | 19 | 201 | 1628 | 5.1 | 21 | 5.3 | 15.4 | 3.0 |
| Average | 0.585 | 11.6 | 189.7 | 1473.8 | 6.08 | 18 | 4.02 | 9.01 | 3.43 |

The computational time is fairly low which is another very attractive advantage of our proposed method. Let's analyze the reason why our proposed

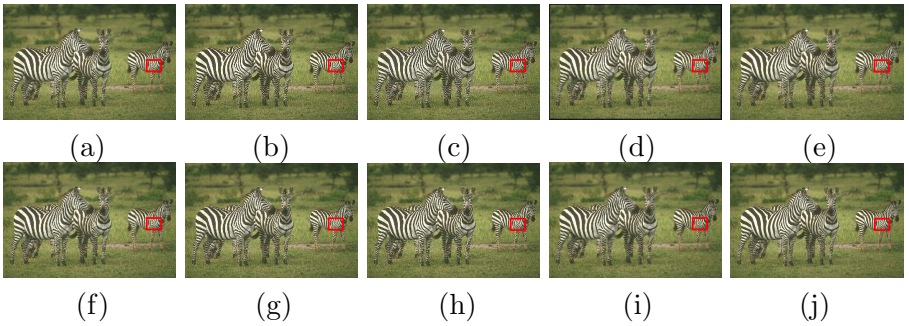



Figure 9: ZebrasRGB: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

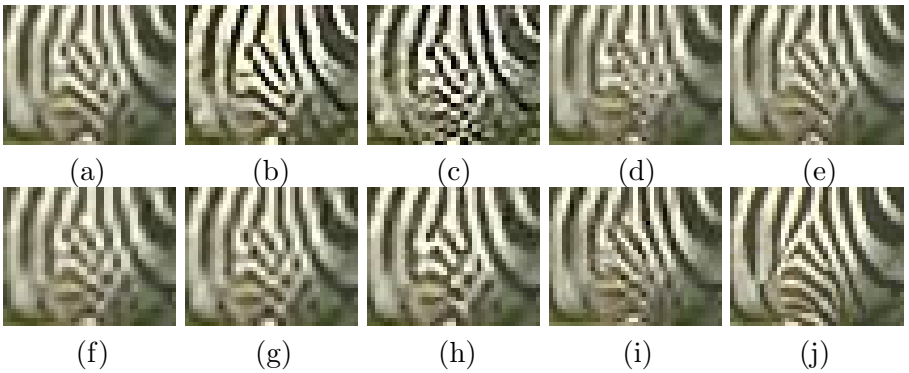



Figure 10: Portions of image "ZebrasRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

algorithm has such a low time cost. For a $m \times n$ image, it has $(m-2) \times (n-2)$ interpolation units. From the construction of interpolation function, the region division of image and the constraint of interpolation pixels, we can know that these calculation processes are mostly of simple algebraic operations. The time complexity of the proposed method is O(1) for an interpolation unit. So the total additional computational complexity is only $O((m-2) \times (n-2))$. Thus the computational complexity of the proposed method, Bicubic and the Liu's method are in the same order.

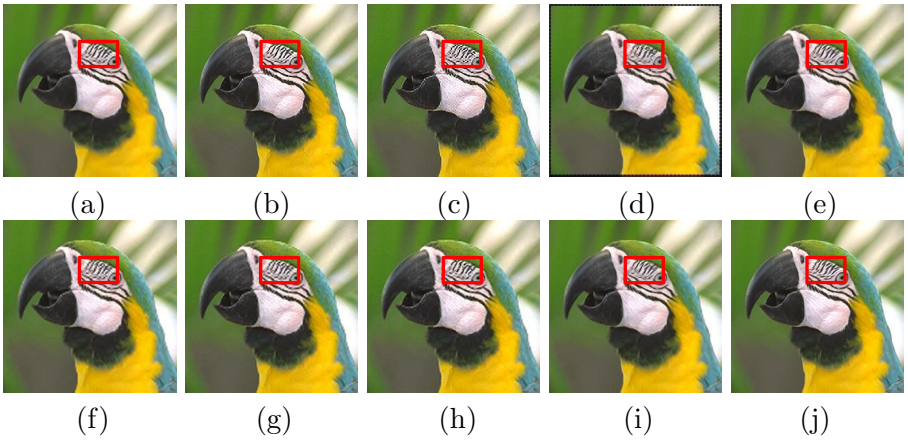



Figure 11: ParrotsRGB: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

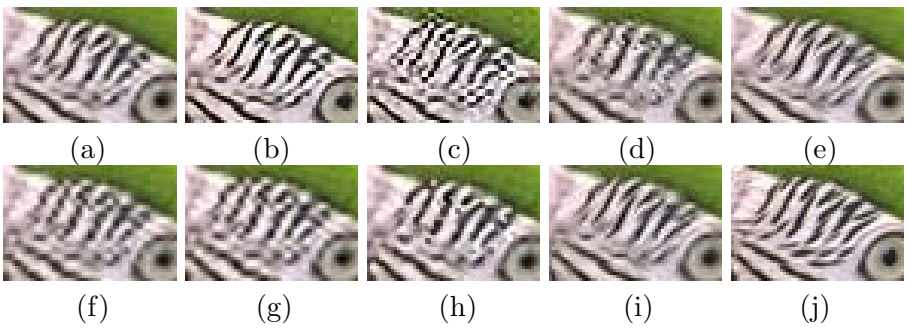



Figure 12: Portions of image "ParrotsRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

In table 3, we demonstrate the runtime of the proposed method and other methods. All the experiments are executed on a PC with an Intel Core i3-2328 2.20 GHz CPU and 2 GB of RAM. All these methods are implemented in Matlab. From the average run time, the speed of the proposed method is about 3.5, 55, 429, 5, 2.6 times faster than that of the DFDF, RSAI, NARM, Lee's, SRCNN method. The computational complexity of

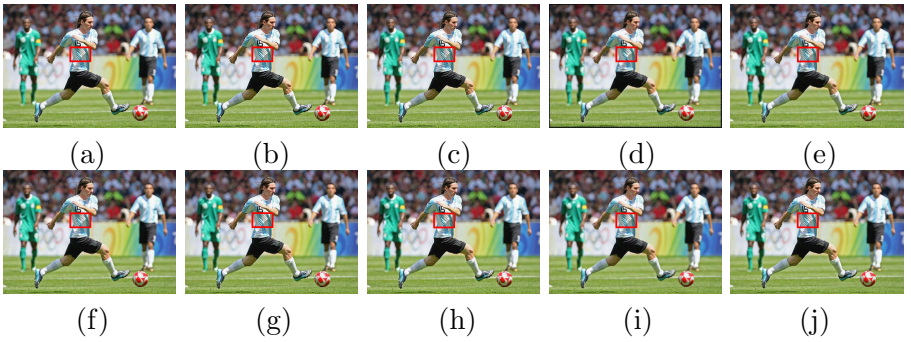



Figure 13: SoccerRGB: (a) NARM; (b) SRCNN;(c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

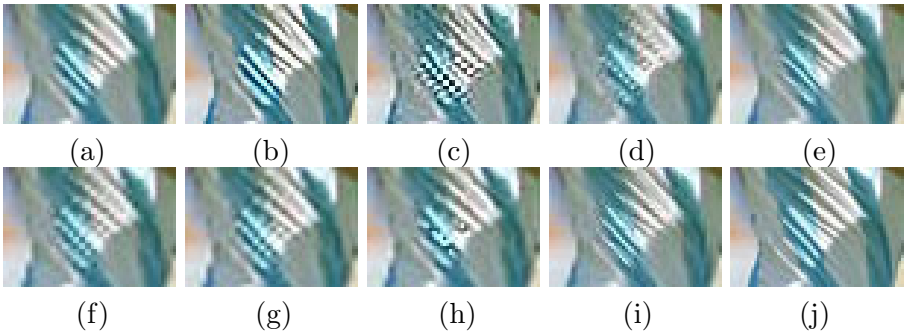



Figure 14: Portions of image "SoccerRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

the proposed method, Liu's and Bicubic are in the same order. Our approach achieved the best interpolation performance at expense of relative low computational complexity.

The proposed algorithm is tested from three aspects: objective data, visual effect and time complexity. Compared with the other methods, the proposed method can effectively maintain the texture detail and edge structure of the image, provides higher objective data and has lower time complexity.

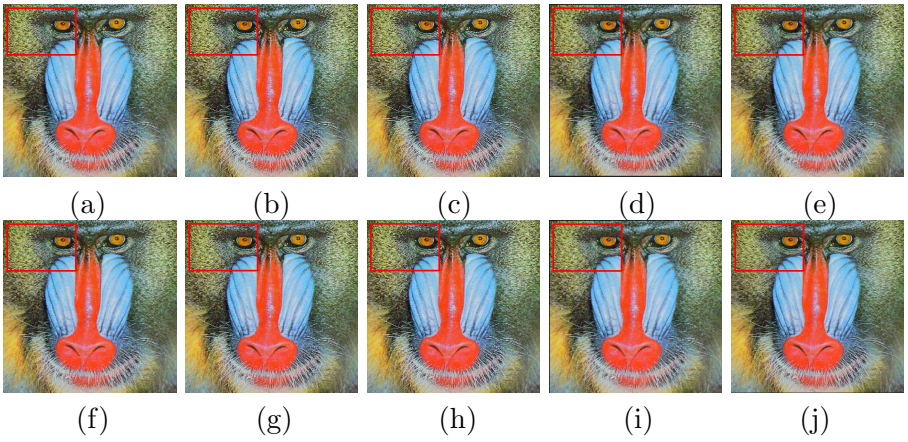



Figure 15: BaboonRGB: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

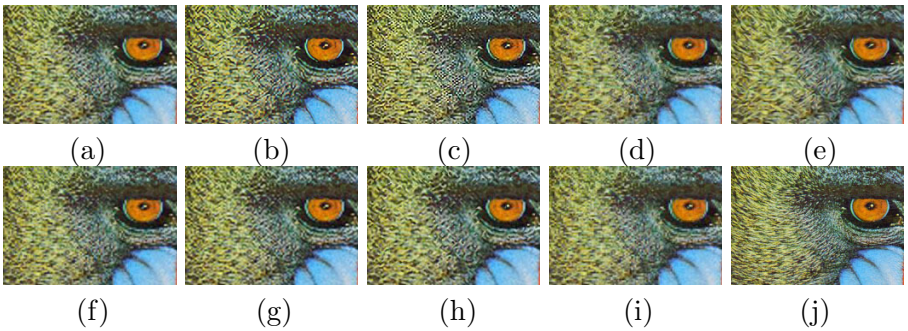



Figure 16: Portions of image "BaboonRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

## 5. Conclusion

We propose a texture feature preserving image interpolation algorithm via gradient constraint. Firstly, based on the proposed rational interpolation model, the isolines method is applied to image regional division. The image is divided into smooth regions and non-smooth regions. Secondly, the smooth

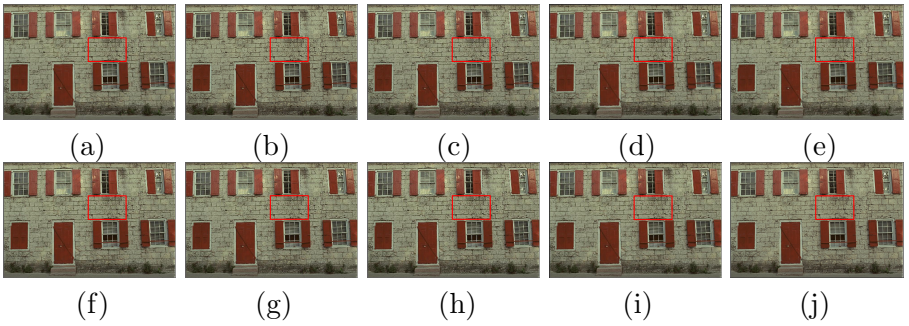



Figure 17: WallRGB: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

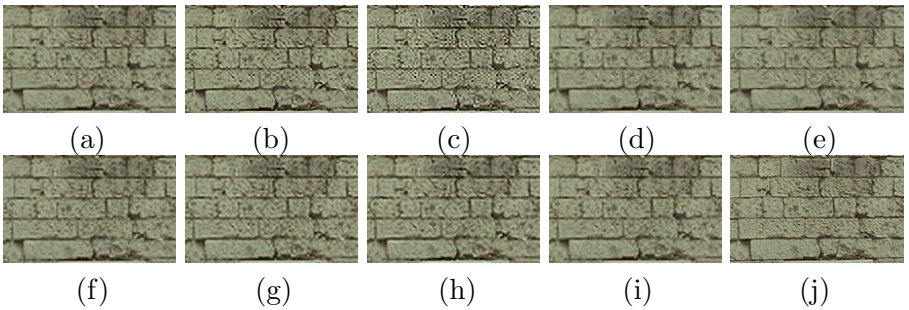



Figure 18: Portions of image "WallRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

region is interpolated by polynomial model, and the non-smooth region is interpolated by rational model. Finally, the texture direction is determined according to the image gradient. The pixel values of the interpolated points are calculated by convolving the matrix of the weight coefficients of the different texture direction interpolation units, and the objective function minimization problem based on the texture direction is solved. Experimental results show that the proposed algorithm achieves good competitive performance compared with the state-of-the-art interpolation algorithms, especially in preserving image details and structure of edge.

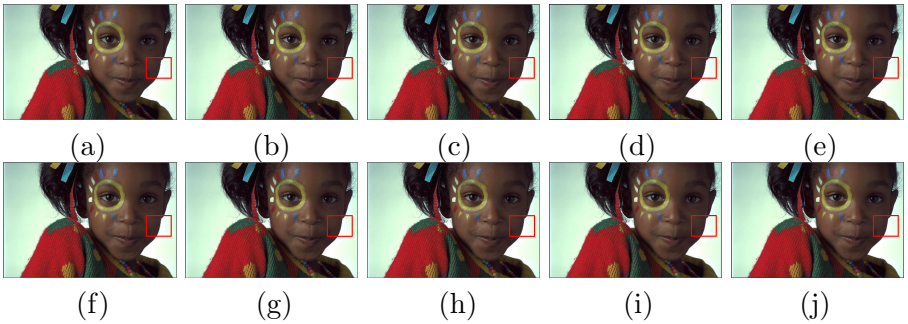



Figure 19: GirlRGB: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

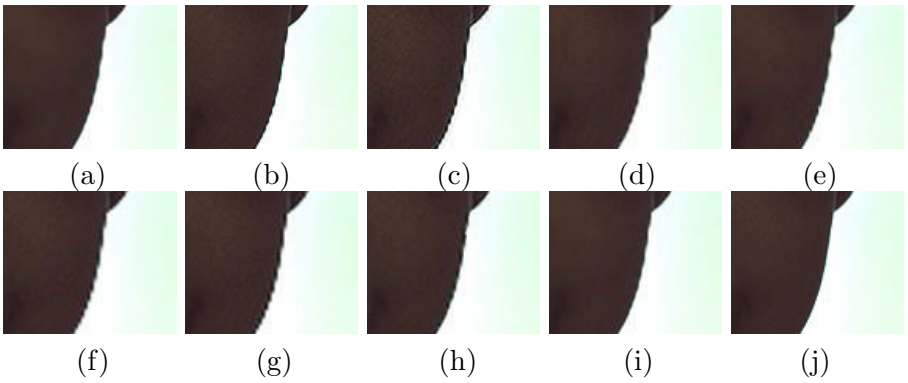



Figure 20: Portions of image "GirlRGB": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

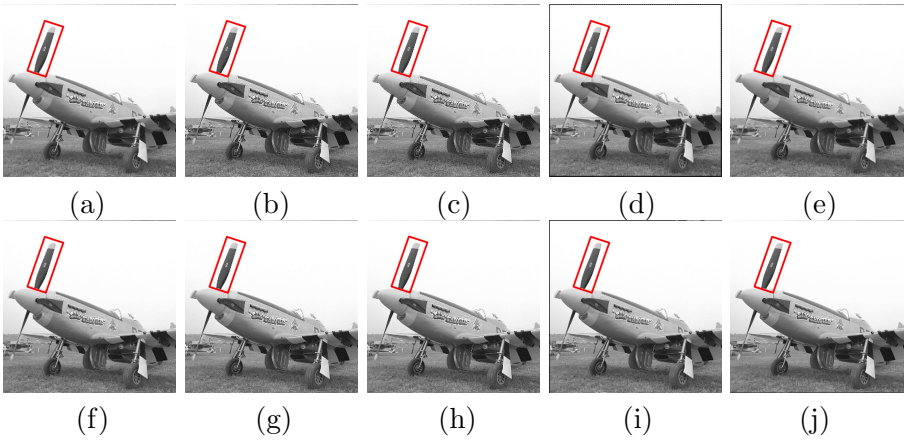



Figure 21: Airplane: (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

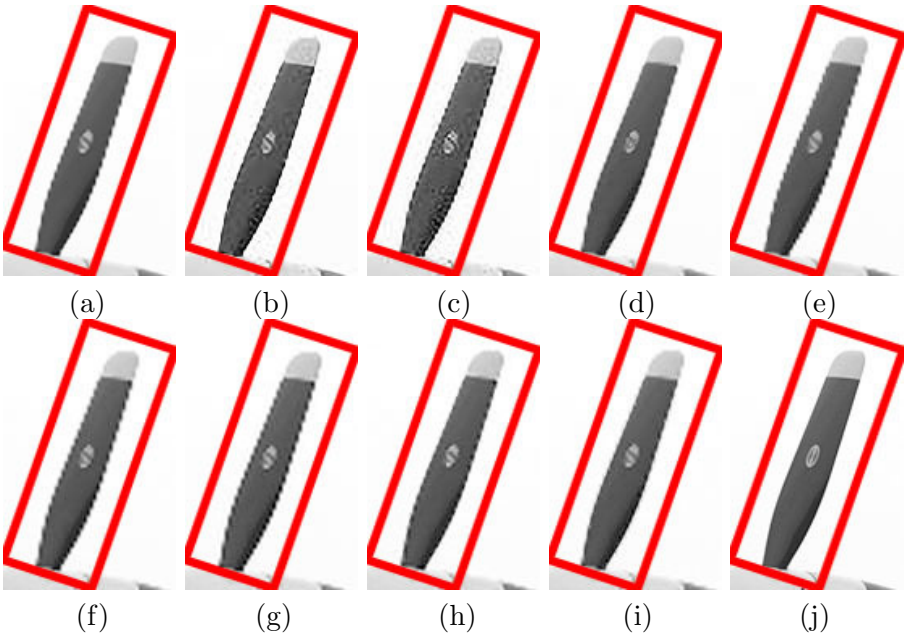



Figure 22: Portions of image "Airplane": (a) NARM; (b) SRCNN; (c) CSCN; (d) DFDF; (e) RSAI; (f) BICUBIC; (g) Liu's; (h) Lee's; (i) our approach; (j) ground-truth image.

## References


[1] R. Key, *Cubic convolution interpolation for digital image processing*, IEEE Transactions on Acoustics Speech and Siginal Processing **29** (2003), no. 6, 1153–1160.

[2] H. Hou and H. Andrews, *Cubic splines for image interpolation and digital filtering*, IEEE Transactions on Acoustics Speech and Siginal Processing **26** (1978), no. 6, 508–517.

[3] A. Jain, *Fundamentals of digital image processing*, Prentice Hall, 1989.

[4] S. Carrato and L. Tenze, *A high quality* 2 *x image interpolatior*, IEEE Signal Processing Letters **7** (2000), no. 6, 132–134.

[5] X. Li and M. Orchard, *New edge-directed interpolation*, IEEE Transactions on Image Processing **10** (2001), no. 10, 1521–1527.

[6] L. Zhang and X. Wu, *An edge-guided image interpolation algorithm via directional filtering and data fusion*, IEEE Transactions on Image Processing **15** (2006), no. 8, 2226–2238.

[7] X. Zhang and X. Wu, *Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation*, IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society **17** (2008), no. 6, 887–896.

[8] K. W. Hung and W. C. Siu, *Robust soft-decision interpolation using weighted least square*, IEEE Transactions on Image Processing **21** (2012), no. 3, 1061–1069.

[9] W. Dong, L. Zhang, R. Lukac, and G. Shi, *Sparse representation based image interpolation with nonlocal autoregressive modeling*, IEEE Transactions on Image Processing **22** (2013), no. 4, 1382–1394.

[10] Z. W. Wang, D. Liu, J. C. Yang, W. Han, and T. Huang, *Deep networks for image super-resolution with sparse prior*, IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[11] C. Dong, C. C. Loy, K. He, and X. O. Tang, *Image super-resolution using deep convolutional networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016), no. 2, 295–307.

[12] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Morel, *Single-image super-resolution via linear mapping of interpolated self-examples*, IEEE Transactions on Image Processing **23** (2014), no. 12, 5334–5347.

[13] Y. Zhang, C. Zhang, J. Chi, and R. Zhang, *An algorithm for enlarged image enhancement*, ICIC Express Letters **3** (2009), no. 3, 669-674.

[14] Q. Duan, H. Zhang, Y. Zhang, and E. H. Twizell, *Bounded property and point control of a bivariate rational interpolating surface*, Computers Mathematics with Applications, **52** (2006), no. 6, 975–984.

[15] Y. Zhang, F. Bao, C. Zhang, and D. Qi, *Local shape control of a bivariate rational interpolation surface with mixing conditions*, Proceedings of the 8th International Symposium on Voronoi Diagrams in Science and Engineering, Los Alamitos: IEEE Computer Society Press (2011), 200–205.

[16] Q. Sun, F. Bao, Y. Zhang, and Q. Duan, *A bivariate rational interpolation based on scattered data on parallel lines*, Journal of Visual Communication and Image Representation **24** (2013), no. 1, 75–80.

[17] M. Sheng, *A new method by rational interpolation for image zooming*, Computer Knowledge and Technology **6** (2010), no. 15, 4019–4022.

[18] M. Hu and J. Tan, *Adaptive osculatory rational interpolation for image processing*, Journal of Computational and Applied Mathematics **195** (2006), no. 1-2, 46–53.

[19] Y. Liu, Y. Zhang, Q. Guo, and C. Zhang, *Image interpolation based on weighted and blended rational function*, Lecture Notes in Computer Science, Heidelberg:Springer (2014), no. 9009, 78–88.

[20] Q. Zhao, M. Hu, and J. Tan, *Many-knot splines technique for image interpolation*, Journal of Image and Graphics **11** (2006), no. 5, 667–671.

[21] Y. Wen and Y. Liu, *An isoline generation algorithm based on delaunay*, International Corference on Computer Engineering and Technology(ICCET) **7** (2010), no. 7, 173–176.

[22] W. Burger and M.J. Burge, *Edges and contours*, Undergraduate Topics in Computer Science (2009), 1–26.

[23] S. J. Lee, M. C. Kang, K. H. Uhm, and S. J. Ko, *An edge-guided image interpolation method using taylor series approximation*, IEEE Transactions on Consumer Electronics **62** (2016), no. 2, 159–165.

Department of Computer Science and Technology
Shandong University of Finance and Economics

Jinan, P.C. 250014, China
*E-mail address*: hongweidumr@163.com

Department of Computer Science and Technology
Shandong University of Finance and Economics
Jinan, P.C. 250014, China
*E-mail address*: yfzhang@sdufe.edu.cn

Department of Mathematica, Shandong University
Jinan, P.C. 250100, China
*E-mail address*: fxbao@sdu.edu.cn

Department of Computer Science and Technology
Shandong University of Finance and Economics
Jinan, P.C. 250014, China
*E-mail address*: 13220931629@163.com

Department of Computer Science and Technology
Shandong University, Jinan, P.C. 250101, China
*E-mail address*: czhang@sdu.edu.cn